

Acoustic watermark server effectiveness

Z. Piotrowski & P. Gajewski

*Telecommunications Institute, Faculty of Electronics,
Military University of Technology, Poland*

Abstract

This paper refers to the technology of the acoustic watermark Internet server, which distributes digital sound together with additional inaudible hidden data. We describe the main architecture of this server (software modules) as well as the main features of this system. The characteristic specification referring to the implemented watermarking server's engine is also presented. Data hiding is one of the most important technologies of Digital Rights Management dedicated for digital multimedia systems. Using hidden signals just below original host level and a perceptual model we can create another digital space for very important data. This data represents the index value for multimedia identification purposes (ID). In this paper we also describe experimental results using the internet infrastructure and dedicated algorithm for audio content fingerprinting. The implemented algorithm operates at the frequency domain and uses orthogonally distributed frequency components allocated along the spectrum bins. We also computed several metrics for a fixed number of audio tracks to show the watermark server and algorithm effectiveness. The following metrics were computed: the time of signal processing for watermark embedding and decoding, the Signal-to-Mask Ratio (SMR), the Mean Square Error (MSE), the Normalised Mean Square Error (NMSE), the Peak Signal to Noise Ratio (PSNR) and the Audio Fidelity (AF).

Keywords: data hiding, watermarking, watermarked sound metrics, digital rights management.

1 Introduction

Nowadays, in multimedia distribution systems the important problem is protection against loss of intellectual work by, e.g., the illegal copying and distribution of digital sound without paying the author or producer. This problem



has been increasing dramatically for several years, thus producers and artists had to decrease their royalties from records to have competitive prices per single CD or DVD record in comparison to the so-called digital pirates. The problem is also important to young music producers and composers. They sometimes produce, e.g., well recognised sound tracks – various kinds of music: hip-hop, house, pop etc. in their “home” studios and after the time consuming creation and development sound track process they just simply open the music to the public for free. Our proposal is an internet system for digitally signing the music. This is simply to mark the sound track with an additional imperceptible signature [3] using an acoustics internet server. Then, after this stage, the so-called watermarked sound tracks are made available to public. This method has many advantages: open access to all users all over the world and individual signature allocation only to the dedicated user as well, among other things.

2 The main system architecture

The system architecture is presented in Figure 1.

The system consists of an internet server with a build-in database. All users’ terminals are connected to the internet. The server distributes both private as well as public signatures (ID), thus the user can allocate proper meta-data to these signatures. The main web page is shown in Figure 2.

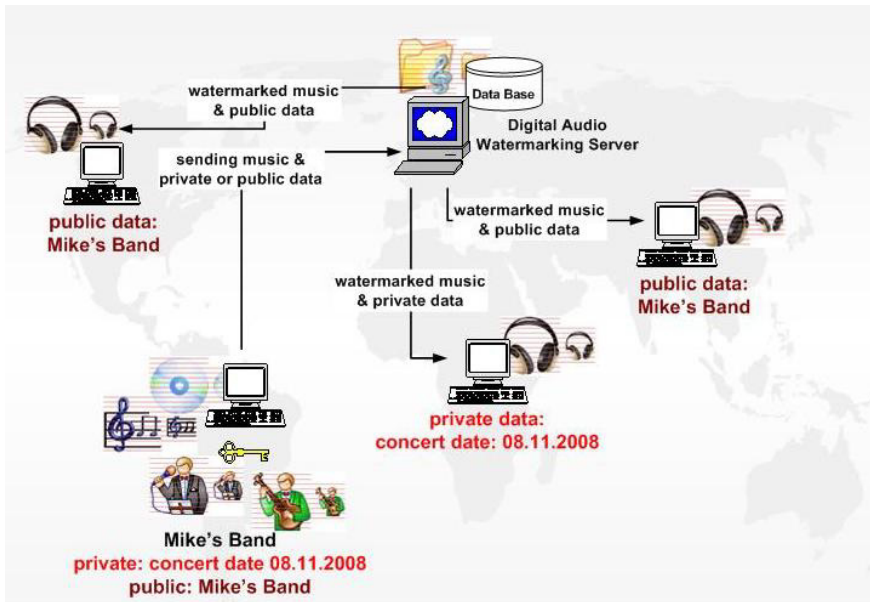


Figure 1: The main system architecture.

The main features of the acoustic watermark internet server are:

- Watermark embedding. (The watermark is a dedicated acoustic signal that is added to the original host signal and is completely inaudible at host signal presence. The watermark represents the digital signature as an integral number.)
- Watermark decoding.
- Sound track watermarking using a signature (the signature value is presented only to the logged user).
- Only a logged user can decode the sound track.
- The server has a built-in algorithm based on frequency domain processing to embed the watermark.

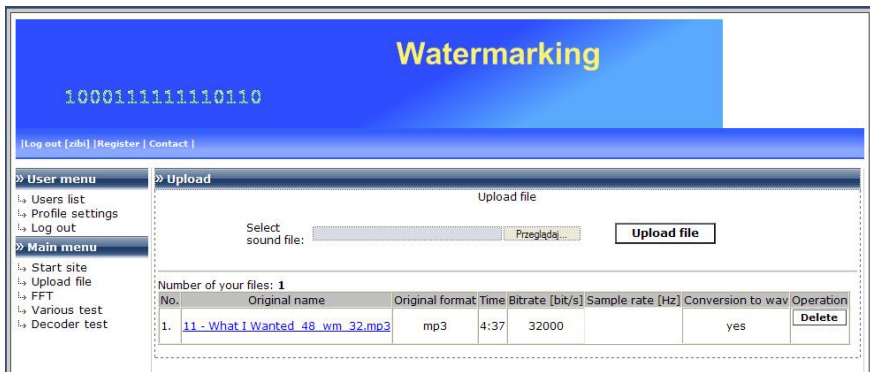


Figure 2: The main web page of the acoustic watermark server.

2.1 Software modules

The developed system is based on a professional database used PostgreSQL technology and PHP scripts, as well as a watermarking engine. The watermarking engine ensures an effective watermark signal is embedding into the original host sound track. The sound track is stored in the database as a *.wav format file.

2.1.1 Database implementation

The base language to communicate with database internet systems is SQL [1]. SQL is a structural language used to save, send and generally for data management purposes in relational database systems. Taking into consideration the project character, a database management system was chosen based on an open source licence server. The basic servers used for these purposes are: MySQL, Firebird and PostgreSQL. We predict very large data numbers, thus the relative most optimal server was the Postgre SQL server. PostgreSQL [2] is an object-relational open-source system compiled to the most popular operational systems e.g. Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) and Windows.



The main table *TSignatures* implemented in the database structure is presented in Figure 3.

The Watermark signature is saved as digital data to the *TSignatures* table. The signature number (“*IW field*”) is also the table main key, which auto increments in successive lines with the signature data. The “*Status*” field allows one to save information about signatures used. The Fields “*IDSystem*” and “*IDUser*” are the foreign keys with the data coding system and users.

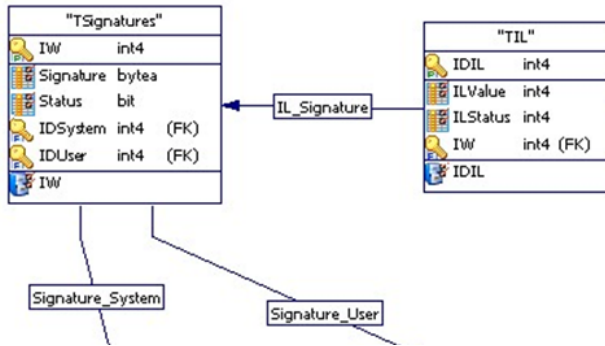


Figure 3: The main table *TSignatures* implemented in the database system.

2.1.2 PHP scripts

The project was written in PHP language in the 5.2 PHP interpreter version using object oriented programming and a MVC model (model-view-controller). Scripts work in two basic modes:

- server-side scripts (response to user action – controller, and content generation – view);
- CLI scripts (command line interface) sound signal processing control.

2.1.2.1 Server-side scripts

The object model requires one to divide programme code into classes to collect the object definition in one file. There are several implemented classes:

- *Main* – the main class controls all project processes. This class is responsible for session initialization, database connection and cookie reading. The two main methods are: *makeAction()* – controls data processing transmitted using forms and *registerFlags()*, responsible for sending required data to view.
- *User* – this class represents the user of the web server. The user can be logged and can also be a guest. This class allows for user registration, registration confirmation and the login process.
- *userAdmin* – this class is similar to the user class, but has methods allowing for user management.
- *files* – this class receives transmitted files from the user, allocates them in a secure manner and is responsible for processing the files by CLI scripts initialization.

- *coder* – this class represents the coding and decoding system.
- *cms* – this class manages the web content.

Additionally there are other, external classes using a GNU GPL licence:

- *phpMailer* – allows for electronic mail sending using external SMTP servers.
- *xajax* – library for quick integration of the AJAX technique with PHP mechanisms.
- *PHPTal* – forms TAL (Template Attributes Language) implementation based on the XML language.
- *getId3* – this class allows for quick file header decoding and obtaining information about the file.
- *reCaptcha* – this class is connected with the recaptcha.org service, using a human recognition module named Completely Automated Public Turing Test to tell Computers and Human Apart.

2.1.2.2 CLI scripts

CLI scripts control the time consuming processes. There are time restrictions for PHP script execution, typically 30 seconds, thus time consuming operations must be executed using CLI scripts. There are many benefits and positive features of using CLI scripts:

- faster view generation (web page generation) for the end user;
- easy management of the dedicated processes by the administrator;
- robustness against random connection braking using a web browser;
- possible process moving to another machine;
- hardware reallocation to the processes.

The client browser has the possibility of communicating with the web service using server-side scripts only.

3 Watermarking engine

The watermarking engine consists of a compiled to binary form watermark coder and decoder. A frequency domain based algorithm was used for watermarking embedding and decoding [2]. The implemented algorithm operates at the frequency domain [4] and uses orthogonally distributed frequency components allocated along the spectrum bins. The psychoacoustic correction method for watermarking signal shaping was used. We computed the basic characteristics of this algorithm using input and output signals; the input is the host signal and the output is the watermarked signal. Each of the frequency components carries its amplitude information in about one bit, thus the more frequency components the more output data payload. Figure 4 presents the watermark signal as an orthogonally distributed frequency component.

4 Experimental results

Based on the compiled version of the watermark algorithm, we compute several signals metrics. In Figure 5 the basic watermarked signal metrics are presented.



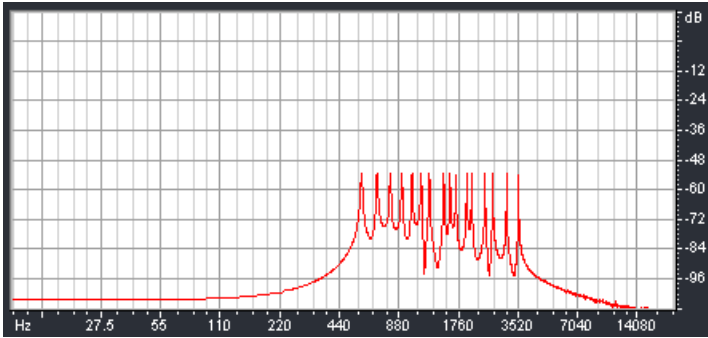


Figure 4: Output watermark signal before the embedding process.

	P_{orig}	P_{zcorr}	P_{out}	SMR	MSE	NMSE	PSNR	AF
001_RS	44,889	25,175	44,938	19,714	-39,051	-19,714	-27,388	0,9893
002_RS	44,987	25,114	45,033	19,873	-39,112	-19,873	-27,507	0,9897
003_RS	43,526	22,881	43,563	20,645	-41,345	-20,645	-24,498	0,9914
004_RS	43,747	21,922	43,775	21,826	-42,304	-21,826	-27,315	0,9934
005_RS	43,071	20,605	43,095	22,466	-43,621	-22,466	-28,075	0,9943
006_RS	43,499	18,653	43,513	24,846	-45,573	-24,846	-23,36	0,9967
007_RS	45,158	24,012	45,192	21,146	-40,213	-21,146	-25,497	0,9923
008_RS	44,021	23,043	44,057	20,978	-41,183	-20,978	-25,001	0,992
009_RS	43,526	21,953	43,558	21,574	-42,273	-21,574	-25,966	0,993
010_RS	32,1	4,2223	32,108	27,878	-60,004	-27,878	-19,781	0,9984
011_RS	45,459	22,964	45,483	22,496	-41,262	-22,496	-25,332	0,9944
012_RS	43,528	21,238	43,553	22,289	-42,988	-22,289	-23,85	0,9941
013_RS	49,422	28,565	49,458	20,856	-35,661	-20,856	-30,415	0,9918
014_RS	48,845	28,216	48,882	20,629	-36,01	-20,629	-30,881	0,9913
015_RS	46,985	26,039	47,021	20,946	-38,187	-20,946	-28,733	0,992
016_RS	50,781	29,341	50,812	21,44	-34,885	-21,44	-31,504	0,9928
017_RS	46,312	22,547	46,328	23,765	-41,678	-23,765	-25,451	0,9958
018_RS	49,215	29,07	49,258	20,145	-35,156	-20,145	-30,466	0,9903
019_RS	39,673	19,97	39,722	19,703	-44,256	-19,703	-25,918	0,9893
020_RS	33,922	13,154	33,957	20,767	-51,072	-20,767	-26,974	0,9916
021_RS	35,987	13,025	36,01	22,962	-51,201	-22,962	-22,147	0,9949
022_RS	44,375	24,42	44,42	19,956	-39,806	-19,956	-29,146	0,9899
023_RS	41,967	20,66	41,999	21,306	-43,566	-21,306	-24,425	0,9926
024_RS	42,066	20,291	42,096	21,775	-43,935	-21,775	-24,976	0,9934
025_RS	42,387	18,777	42,405	23,61	-45,449	-23,61	-26,354	0,9956
mean	43,57792	21,83429	43,60944	21,74364	-42,3916	-21,7436	-26,4384	0,992812
min	32,1	4,2223	32,108	19,703	-60,004	-27,878	-31,504	0,9893
max	50,781	29,341	50,812	27,878	-34,885	-19,703	-19,781	0,9984
std	4,470551	5,559624	4,474195	1,840733	5,559685	1,840733	2,816093	0,002346

Figure 5: Basic metric of the watermarked signal.

We use following metrics to compute the basic parameters of the watermarked signal:

P_{orig} - power [dB] of the host acoustic signal

P_{zcorr} - power [dB] of the corrected watermark signal (corrected means – shaped according Just Noticeable Level (JND=0))

P_{out} - power [dB] of the output acoustic signal with watermark signal

$$SNR = \frac{\sum A_n^2}{\sum_n (A_n - A'_n)^2} \quad - \quad \text{Signal-to-Noise Ratio}$$

$$MSE = \frac{1}{N} \sum |A_n - A'_n| \quad - \quad \text{Mean Square Error}$$



$$\begin{aligned}
 NMSE &= \sum (A_n - A'_n)^2 / \sum_n A_n^2 & - & \text{Normalized Mean Square Error} \\
 PSNR &= N \max_n A_n^2 / \sum (A_n - A'_n)^2 & - & \text{Peak Signal to Noise Ratio} \\
 AF &= 1 - \sum (A_n - A'_n)^2 / \sum A_n^2 & - & \text{Audio Fidelity}
 \end{aligned}$$

We compute the server average processing time of the *.wav format file. In Table 2 we show the results of this experiment for the fixed duration time of the *.wav file (sampling frequency 44100 Hz and 16 bits/sample). Software tool versions are presented in Table 1.

Table 1: Software tools installed on the experimental server.

no.	tool	version
1	OpenBSD	4.3
2	PostgreSQL	8.2.6
3	PHP 5	5.2.5
4	Perl	5.8.8
5	Apache	1.3
6	cgi	1.1
6	SSL	yes

Table 2: Average *.wav file processing time by server.

duration time [s]	mono (*.wav) [s]	stereo (*.wav) [s]
5	7.5	8.6
10	9.3	10.8
30	13.8	15.2

5 Conclusions

An acoustic watermark server based on a built-in PostgreSQL database and a watermarking engine (compiled to binary code) can be very helpful for Digital Rights Management as a basic tool for acoustic hidden fingerprinting (the unique ID signature is represented by the watermark signal). The relatively high processing time for *.wav standard files can be decreased by better server configuration, as well as software optimization. The watermark signal is inaudible in the presence of the host signal.

References

- [1] ISO/IEC 9075-1:2008, Information technology – Database languages – SQL - Part1: Framework (SQL/Framework).
- [2] Piotrowski Z., Gajewski P.: *Novel method for watermarking system operating on the HF and VHF radio links*, Computational Methods and Experimental Measurements XIII, CMEM XIII, , Southampton, Boston, WIT Press, pp.791-800, 2007



- [3] Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems; Recommendation ITU-R BS.1116.
- [4] Richard G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall PTR Publication, 2004

